

# SigmaForge

## LLM-Assisted Sigma Rule Engineer for Real SIEMs

From threat intel to validated Sigma rules and SIEM queries in minutes

Track

**Cybersecurity & Infrastructure Protection**

Hackathon

**Apart Research Defensive Acceleration**

Team: [Your Name(s)]

# The Detection Bottleneck in AI-Accelerated Attacks

## AI MAKES OFFENSE FASTER

Auto-mutated malware, stolen prompts, rapid TTP changes

## DEFENDERS RELY ON SIGMA RULES

SIEM queries catch attacks in logs

## WRITING RULES IS HARD

Slow: hours per rule

Expert-heavy: senior detection engineers

In short supply: global cyber skills gap

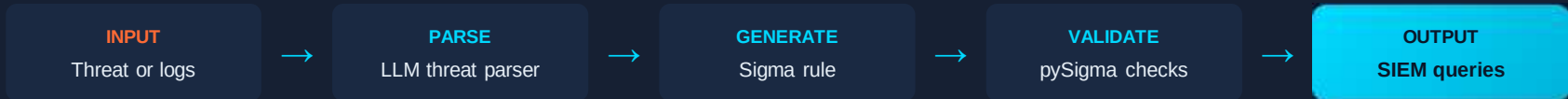
## THE RESULT

New attack known → days/weeks before orgs have a working rule

We need a faster way to translate threat intel into high-quality detections

# SigmaForge Pipeline

FROM "WE KNOW ABOUT THIS ATTACK" → "WE ARE DETECTING IT"



## PASTE

Plain-English threat description  
Representative log lines

## SIGMAFORGE EXTRACTS

Logsource and fields  
ATT&CK techniques  
Validates and compiles to SIEM queries

# Feature Set Built for Real SOC Workflows

## THREAT → JSON

LLM extracts logsource, fields, ATT&CK techniques

## RULE GENERATION

Produces vendor-agnostic Sigma YAML rules

## SIEM CONVERSION

Uses pySigma to convert rules into:

Splunk SPL queries (Windows pipeline)

Elasticsearch / Lucene queries

## ENHANCED VALIDATION

Quality checks for:

Missing tags / ATT&CK IDs / descriptions

Risky wildcards and false positives

## QUICK LOG SANITY CHECK

Paste logs and see which lines match the rule — instant feedback

Expandable sections ready to paste directly into your SIEM

# Beyond "ChatGPT, Write Me a Rule"

## 1. DEEP SIGMA INTEGRATION

- Uses official pySigma parser & backends
- Converts to real Splunk / Elasticsearch queries
- Validates with community tooling

## 2. PRODUCTIVE CONSTRAINTS

- Dedicated prompts for each step
- Threat parsing → JSON only
- No free-form chat, fixed pipeline

## 3. QUALITY & SAFETY LAYER

- pySigma validation + static checks
- Catches noisy rules before deployment
- Defensive-only: no exploit code

## 4. ACTUAL MINI-EVALUATION

- Tested on multiple scenarios:
  - SSH brute force
  - PowerShell downloads
  - Web shell uploads
  - Impossible travel

**Not a wrapper — a production pipeline that accelerates the defensive workflow**

# Why SigmaForge Advances Defensive Acceleration

## DIRECTLY ADDRESSES AI-ENABLED THREATS

Attackers already use automation & AI to evolve TTPs quickly. SigmaForge accelerates the defensive side — turning new intel into detections.

## STRENGTHENS THE BOTTLENECK: DETECTION ENGINEERING

Juniors can draft rules; seniors focus on reviewing & tuning. Scales expertise across orgs using open standards.

## REAL, DEPLOYABLE PROTOTYPE

Working Streamlit app

Runs in GitHub Codespaces with one click

Open-source codebase built on real industry tooling

## PATH TO IMPACT AFTER THE HACKATHON

Could plug into existing SOC workflows as:

A detection-as-code assistant

A Sigma rule doctor over the public Sigma repo

A CI step for validating rules before deployment

Defensive acceleration = shipping tools that defenders can realistically adopt tomorrow

# Evaluation, Limitations & Next Steps

## WHAT WE EVALUATED

### 4 scenarios with synthetic logs:

SSH brute force → success

Suspicious PowerShell download

Web shell upload

Impossible travel logins

### For each, we checked:

Does the rule match malicious lines?

How many benign lines are flagged?

## LIMITATIONS & FUTURE WORK

Still reliant on LLM correctness

Needs human in the loop

Current log tests are small-scale

### Next steps:

Integrate with real log datasets

Add "environment profiles" for SIEMs

CI integration for rule validation

### Long-term:

Extend to Suricata, Zeek backends

Integrate with Blue Team platforms

SigmaForge shows how we can use modern LLMs to accelerate, not replace, human defenders — and turn Sigma + SIEM ecosystems into something teams can keep up to date at AI speed